

# 《C/C++ 学习 指南》

## 第28.1讲：内部类

作者：邵发    QQ群：417024631  
官网：<http://afanihao.cn>

C/C++学习指南    <http://afanihao.cn>

## 定义内部类

把一个类的定义写在另一个类的内部，则称里面的这个类为内部类。例如，Inner就是

```
class AAA
{
public:
    // 定义一个内部类
    class Inner
    {
    public:
        char  name[64];
    };
};
```

## 使用内部类

在使用该内部类时，类名要使用全称 `AAA::Inner`，例如

```
int main()
{
    AAA::Inner a ;
    strcpy(a.name, "AnXin");
    printf("Name: %s \n", a.name);
    return 0;
}
```

除此之外，内部类和普通类没有任何区别。

## 注意事项

**内部类和外部类互相没有特权**

它们不是朋友关系

它们不是父子关系

它们没有任何特殊关系

所以，

外部类无法自由访问内部类，内部类也无法自由访问内部类

在VC中演示。。。

## 内部类的访问限制

如果内部类被protected/private限制，**则该类在外部不可见。**  
(类名被完全隐藏)

例如，

```
class AAA
{
    private:
        class Inner{};
};
int main()
{
    AAA::Inner i; // 编译错误!
    return 0;
}
```

## 内部类的分离式写法

也可以把内部的成员函数定义分离到\*.cpp中实现。。。

```
////////// AAA.cpp //////////
AAA::Inner::Inner()
{
}
void AAA::Inner::Test()
{
    printf("name: %s \n", name);
}
```

加上类名前缀就可以了。。。

## 为什么要用内部类？

即然内部类和普通类并无区别，那为什么要把它写在另一个类的内部呢？

原因：

- (1) 避免名字冲突
- (2) 如果一个类只在模块内部使用，则可以实现类名隐藏

当项目越来越大，避免类名的冲突成为一个现实存在的问题。。。将一些只在模块内部使用的类，写成内部类。

## 为什么要用内部类？

例如，一个视频Video的书签Bookmark表示用户上一次的播放位置。一本小说Novel的书签Bookmark用于表示上一次的页码。

因此，可以定义Video::Bookmark和Novel::Bookmark两个内部类。。。。

当然，你也可以把类名定得长一点来避免重复。。。

Video\_Bookmark Novel\_Bookmark

注：内部类还可以再定义内部类，没有层数限制，但一般就写两层